PAMANTASAN NG LUNGSOD NG MAYNILA
(University of the City of Manila)
Intramuros, Manila

Elective 3

Laboratory Activity No. 1

Image Acquisition and Manipulation

Score

Submitted by:

Cacal, Chad R.
Dela Cruz, Alfonso Rafael C.
Palabay, Joven Carl B.
Suyu, Chester T.

SAT 7:00AM – 4:00PM / CPE 0332.1-1

Date Submitted

07/26/2024

Submitted to:

Engr. Maria Rizette H. Sayo

I. Objectives

This laboratory activity aims to implement the principles and techniques of image acquisition through MATLAB/Octave and open CV using Python

- Acquire the image.
- Rotate the image by 30 degrees.
- Flip the image horizontally.

II. Methods

A. Perform a task given in the presentation

- Copy and paste your MATLAB code

```
% Read the image img = imread('E:\PLM CET SUBJECTS\Digital Image
Processing\flower.jpg');
% Rotate by 45 degrees
rotated_img = imrotate(img, 45);

% Flip horizontally
flipped_img = fliplr(rotated_img);

% Display results
figure(1);
plot(1,1);
imshow(img);
title('Original Image');
figure(2);
plot(1,1);
imshow(rotated_img);
title('Rotated 45°'); figure(3); plot(1,1);
imshow(flipped_img); title('Rotated & Flipped');
```

B. Supplementary Activity

- Write a Python program that will implement the output in Method A.

```
import cv2
#Read the image
img = cv2.imread(r"C:\Users\Chad\OctaveFiles\flower.jpg")
#Rotate by 45 degrees
center_img = (img.shape[1]//2, img.shape[0]//2)
rotation_img = cv2.getRotationMatrix2D(center_img, 30, 1)
rotated_img = cv2.warpAffine(img,
rotation_img,(img.shape[1],img.shape[0]))
#Flip horizontally
flipped_img = cv2.flip(rotated_img,1)
#Display results
cv2.imshow('Original Image', img)
cv2.imshow('Rotated 30°', rotated_img)
cv2.imshow('Rotated & Flipped', flipped_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

C. Results

1. Copy/crop and paste your results. Label each output (Figure1, Figure2, Figure3)

Octave Results

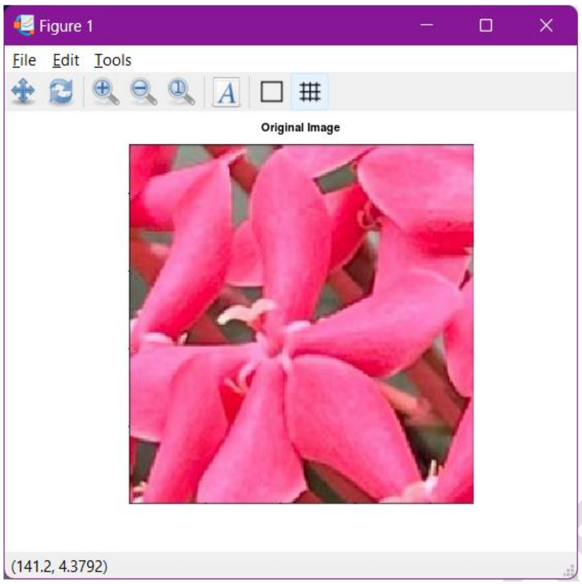

Figure 1. Acquire an Image of a Flower using Octave

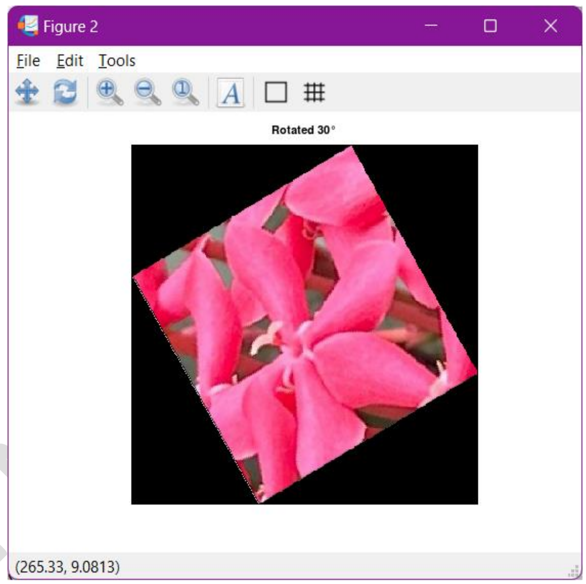

Figure 2. Rotate by 30 degrees in Octave

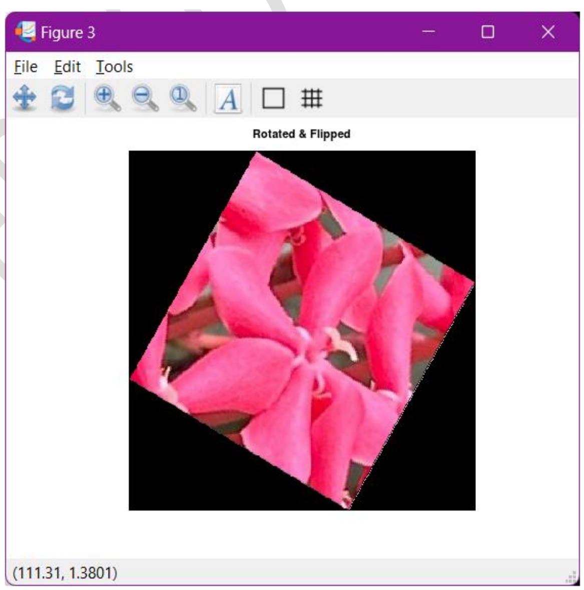

Figure 3. Flip horizontally in Octave

MATLAB Results

Figure 4. Acquire an Image of a Flower using MATLAB

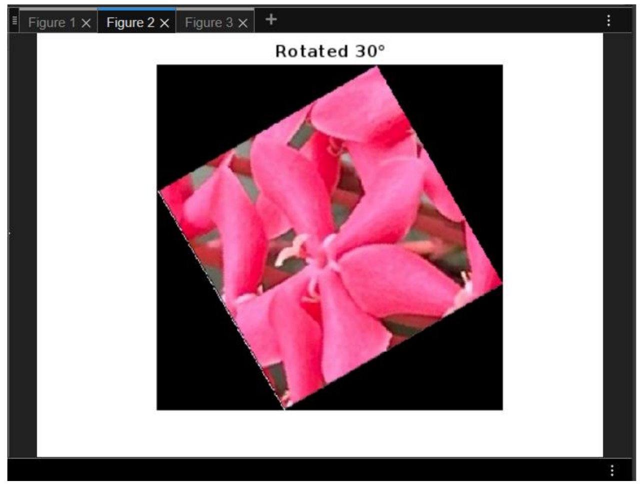

Figure 5. Rotate by 30 degrees in MATLAB

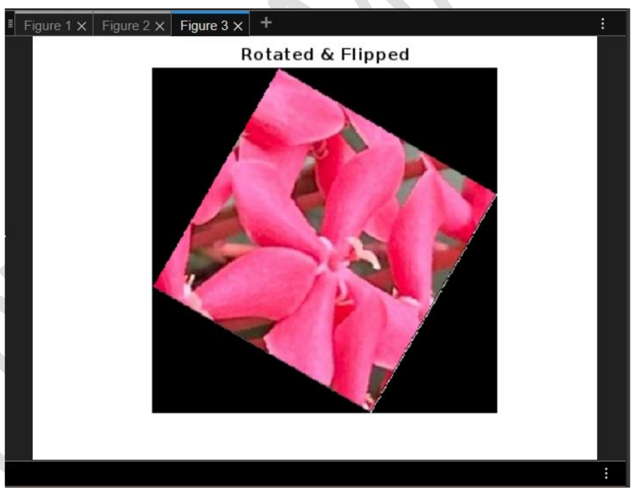

Figure 6. Flipped horizontally in MATLAB

Python/OpenCV Results

Figure 7. Acquire an Image of a Flower using OpenCV Python

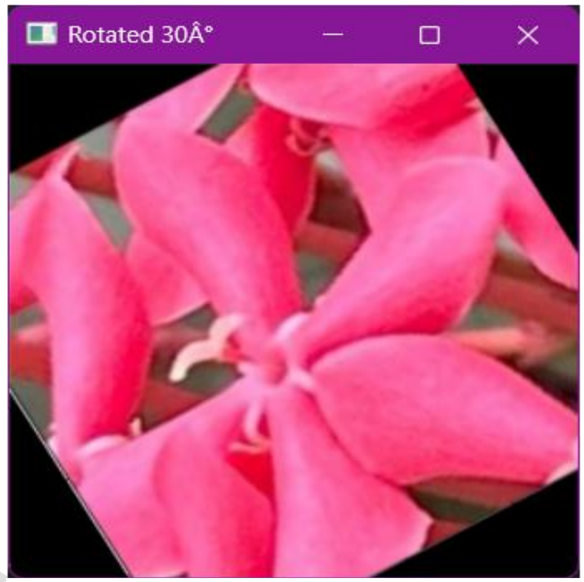

Figure 8. Rotate by 30 degrees in OpenCV Python

Figure 9. Flip horizontally in OpenCV Python

2. Visualize the results, analyze and interpret:

< Discuss the effects of the applied algorithm on the image and its effectiveness in achieving the desired outcome. Handwritten>

Starting with acquiring the image, all of the algorithms used in this activity were able to display the desired outcome. However, when rotating the image by 30 degrees, OpenCV in Python wasn't able to achieve the desired outcome since some parts of the picture were "cropped" unlike the displayed outputs when we used Octave and MATLAB. The unexpected output when we used OpenCV-Python was caused by how cv2.getRotationMatrix2D() function works. The previously mentioned function doesn't consider the changes in the dimension of the picture after the rotation causing some of the picture to be "cropped" when the current dimension of the figure wasn't able to contain all parts of the picture. On the other hand, both Octave and MATLAB were able to achieve the desired outcome. Lastly, when flipping the image horizontally, all of the algorithms used where able to flip the image horizontally however the flipped image when we use OpenCV-Python was also "cropped" due to the rotated image.

IV. Conclusion

In conclusion, the laboratory activity successfully demonstrated the implementation of fundamental principles and techniques of image acquisition and manipulation using MATLAB/Octave and OpenCV with Python. The tasks included acquiring an image, rotating it, and flipping it, showcasing the versatility and power of these tools in handling image processing tasks. Through hands-on practice, we gained practical experience in using MATLAB/Octave for initial image acquisition and leveraging OpenCV with Python for advanced image manipulation. This integration of software platforms enhances the ability to perform comprehensive image processing workflows, crucial for various applications in computer vision and related fields.

PAMANTASAN NG LUNGSOD NG MAYNILA

(University of the City of Manila)
Intramuros, Manila

References

<This is in a separate page>

[1] D.J.D. Sayo. "University of the City of Manila Computer Engineering Department Honor Code," PLM-CpE Departmental Policies, 2020.

codingOtter. (2022, May 8). Image and Bounding Box Rotation Using OpenCV (Python). https://medium.com/@coding-otter/image-and-bounding-box-rotation-using-opencv-python-2def6c39453

Cloudinary. (2024, April 21). Rotating an Image in Python. https://cloudinary.com/guides/image-effects/rotating-an-image-in-python